



Apostila de Introdução à Linguagem HTML

- Disponibilização de Conteúdos na WEB -

Profa. Flávia Pereira de Carvalho

Taquara - RS

Sumário

	<i>Página</i>
1 INTRODUÇÃO À LINGUAGEM HTML	3
2 EDIÇÃO DE DOCUMENTOS HTML	4
3 PUBLICAÇÃO DE DOCUMENTOS NA INTERNET (SITES)	5
4 DOCUMENTO HTML BÁSICO E SEUS COMPONENTES	6
5 SEÇÃO <HEAD>	7
6 SEÇÃO <BODY>	8
<u>6.1 ATRIBUTOS DE <BODY></u>	<u>8</u>
7 FONTES	9
8 CABEÇALHOS	11
<u>8.1 ANINHAMENTO DE CABEÇALHOS</u>	<u>11</u>
<u>8.2 ALINHAMENTO DE CABEÇALHOS</u>	<u>12</u>
9 SEPARADORES	13
<u>9.1 QUEBRA DE LINHA</u>	<u>13</u>
<u>9.2 PARÁGRAFOS</u>	<u>13</u>
10 LINHA HORIZONTAL	15
11 COMENTÁRIOS	15
12 LISTAS EM HTML	16
<u>12.1 LISTAS COM MARCADORES</u>	<u>16</u>
<u>12.2 LISTAS NUMERADAS</u>	<u>17</u>
<u>12.3 LISTAS E SUB-LISTAS</u>	<u>18</u>
13 FORMATAÇÃO DE FRASES	19
14 MARQUEE	19
15 LIGAÇÕES (USO DE LINKS)	20
<u>15.1 ATRIBUTOS</u>	<u>20</u>
<u>15.2 CAMINHOS</u>	<u>21</u>
<u>15.3 INDICADORES</u>	<u>22</u>
16 INSERÇÃO DE IMAGENS	23
<u>16.1 ATRIBUTOS BÁSICOS DE IMAGEM</u>	<u>23</u>
17 MOLDURAS DE IMAGEM	26
18 TABELAS	27
<u>18.1 ELEMENTOS BÁSICOS DE TABELAS</u>	<u>27</u>
<u>18.2 TÍTULOS, LINHAS E ELEMENTOS</u>	<u>27</u>
<u>18.3 UMA TABELA SIMPLES</u>	<u>28</u>
<u>18.4 TÍTULOS COMPREENDENDO MAIS DE UMA COLUNA OU LINHA</u>	<u>28</u>
<u>18.5 TABELAS SEM BORDA</u>	<u>29</u>
<u>18.6 EXTENSÕES DE TABELAS</u>	<u>29</u>
19 FORMULÁRIOS HTML	31
<u>19.1 CRIANDO UM FORM</u>	<u>31</u>
<u>19.2 PARÂMETROS DO FORM</u>	<u>31</u>
<u>19.3 ELEMENTOS DO FORM</u>	<u>32</u>
<u>19.4 TRABALHANDO COM OS DADOS ENVIADOS PELO FORM</u>	<u>35</u>
<u>19.5 COLOCANDO EM PRÁTICA</u>	<u>35</u>

1 Introdução à Linguagem HTML

HTML (*HyperText Markup Language* - Linguagem de Formatação/Marcação de Hipertexto) é fruto do "casamento" dos padrões HyTime e SGML.

HyTime - Hypermedia/Time-based Document Structuring Language: *HyTime* (ISO 10744:1992) - padrão para representação estruturada de hipermídia e informação baseada em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (áudio, vídeo, etc.), conectados por *webs* ou *hyperlinks*. O padrão *HyTime* é independente dos padrões de processamento de texto em geral. Ele fornece a base para a construção de sistemas hipertextos padronizados, consistindo de documentos que aplicam os padrões de maneira particular.

SGML - Standard Generalized Markup Language: Padrão ISO 8879 de formatação de textos: não foi desenvolvido para hipertexto, mas torna-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações. SGML não é padrão aplicado de maneira padronizada: todos os produtos SGML têm seu próprio sistema para traduzir as etiquetas (*tags*) para um particular formatador de texto.

DTD - Document Type Definition: Define as regras de formatação para uma dada classe de documentos. Um DTD ou uma referência para um DTD deve estar contido em qualquer documento conforme o padrão SGML.

Portanto, HTML é definido segundo um DTD de SGML.

Todo documento HTML apresenta elementos entre **parênteses angulares** (< e >). Esses elementos são as **etiquetas** (*tags*) de HTML, que são os comandos de formatação da linguagem. A maioria das etiquetas tem sua correspondente de fechamento, representada com uma "barra" (/):

<etiqueta> texto a ser formatado **</etiqueta>**

Isso é necessário porque as etiquetas servem para definir a formatação de uma porção de texto, e assim marcamos onde começa e onde termina o texto com a formatação especificada por ela.

Alguns elementos são chamados "vazios", pois não marcam uma região de texto, apenas inserem alguma coisa no documento, não havendo a necessidade do fechamento:

<etiqueta>

Todos os elementos podem ter atributos:

<etiqueta atributo1=valor1 atributo2=valor2>...</etiqueta>

HTML é um recurso muito simples e acessível para a produção de documentos. Nesta apostila, será possível aprender uma introdução aos seus elementos.



Dica: Não existem programas em HTML, pois HTML não é uma linguagem de programação, mas de formatação (marcação). Portanto, a rigor, não existem "programadores" de HTML.

2 Edição de Documentos HTML

Existem **Editores HTML** chamados WYSIWYG (*what you see is what you get* - o que você vê é o que você tem). Eles oferecem ambiente de edição com um resultado final das marcações (pois o resultado final depende do *browser*¹ usado para visitar a página). Alguns bastante conhecidos são, por exemplo: FrontPage e Dreamweaver.

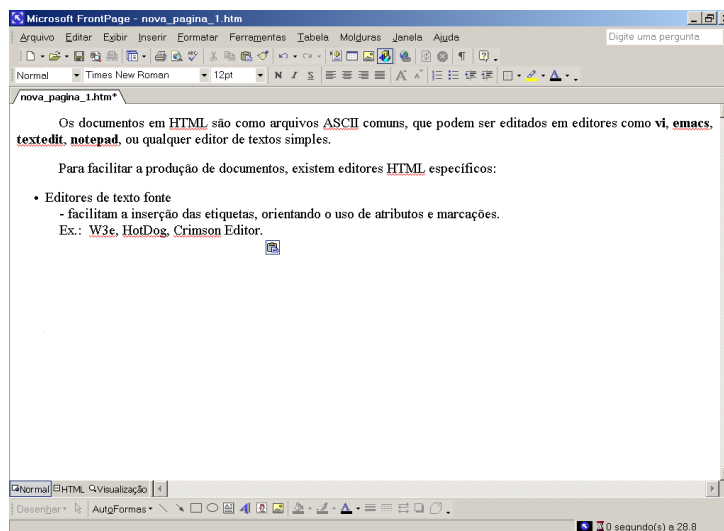


Figura 1 – Tela do FrontPage

Além dos editores específicos para HTML, **Editores de Textos** bastante utilizados, como o Word, entre outros, permitem a exportação de seus documentos próprios para o formato HTML (menu Arquivo, Salvar como, Salvar como Tipo). Um documento HTML, normalmente terá extensão **.html** ou **.htm**.

Existem vários **Editores HTML Online**, como por exemplo: [Online HTML Editor](#), [Mozilla Thimble](#), entre outros.

¹ *Browser*: Navegador Internet. Exemplos: Internet Explorer, Mozilla Firefox, Chrome etc.

3 Publicação de Documentos na Internet (Sites)

Para que uma página esteja permanentemente disponível na Web, ela precisa ter um endereço fixo, **alojada em um servidor**.

Existem vários provedores de espaço (*hosting*) gratuitos e também os provedores de acesso geralmente oferecem espaço para os *sites* de seus assinantes. *Sites* com fins lucrativos geralmente são hospedados em provedores de espaço pagos.

Definida a hospedagem, basta enviar para o provedor os arquivos de seu *site* (via FTP² ou por uma página de envio no próprio provedor de espaço) e suas páginas já estarão disponíveis para visitas no mundo todo. Um programa bastante usado para envio de arquivos de um computador local para um servidor e vice-versa, através do Windows, é o [WinSCP](#).

Em geral, tem-se como arquivo da página inicial, o arquivo chamado **index.html**. Esse é o arquivo *default* (padrão) em um diretório. Ao entrar em uma URL que não especifica um arquivo em um diretório, o *browser* solicita ao servidor o arquivo `index.html`. É por isso que se pode escrever tanto `www.exemplo.com.br` ou `www.exemplo.com.br/index.html`. Se não existir um arquivo chamado `index.html`, o servidor poderá retornar uma mensagem de "acesso proibido à listagem do diretório" ou então retornar a listagem dos arquivos daquele diretório, o que pode não ser o que se deseja. Esse arquivo *default* é definido na configuração do servidor WWW.

² FTP - *File Transfer Protocol*: Protocolo de Transferência de Arquivos.

4 Documento HTML Básico e seus Componentes

A estrutura de um documento HTML apresenta os seguintes componentes básicos:

```
<html>
  <head>
    <meta charset="utf-8">
    <title> Título do Documento </title>
  </head>
  <body>
    textos,
    imagens,
    links, etc...
  </body>
</html>
```

Figura 2 – Componentes Básicos de um Documento HTML

As etiquetas (*tags*) HTML **não são sensíveis à caixa** (a linguagem HTML não é *case sensitive*), ou seja, tanto faz escrever <HTML>, <Html>, <html>, <HtMl>, etc. Não tem diferença entre maiúsculas e minúsculas. Mas, a prática da “boa programação” diz que deve-se usar letras minúsculas no código fonte de páginas web e também de programas em geral.

Os documentos HTML se dividem em seções, apresentadas nos capítulos a seguir.

5 Seção <HEAD>

<HEAD> contém informações sobre o documento. A etiqueta HTML <meta charset="utf-8"> significa que: **meta** é usado para declarar metadados (informações sobre a própria página, como resumo do conteúdo, palavras-chaves, indicações a robôs de busca, entre outras), **charset** é usado para indicar o formato de codificação de caracteres usado no document e **utf-8** é um formato de codificação, ou seja, é usado para mostrar ao navegador o tipo de codificação que será utilizada no site e, assim, funcionar corretamente todas acentuações, por exemplo. O elemento <TITLE> define um título, que é mostrado no alto da janela do *browser*. Exemplo:

```
<head>
  <title> Página do Fulano de Tal </title>
</head>
```

Todo documento web deve ter um título; esse título é referenciado em buscas pela rede, dando uma identidade ao documento. Ao adicionar uma página aos seus Favoritos (*Bookmarks*), o título da página se torna a âncora de atalho para ela. Por isso é sugerido que os títulos dos documentos sejam sugestivos, evitando-se títulos genéricos como "**Introdução**", por exemplo. O título também é bastante significativo para a listagem de uma página nos resultados de pesquisas nos *sites* de busca da Internet.

6 Seção <BODY>

Tudo que estiver contido em <BODY> será mostrado na janela principal do *browser*, sendo apresentado ao leitor. <BODY> pode conter cabeçalhos, parágrafos, listas, tabelas, *links* para outros documentos, imagens, formulários, animações, vídeos, sons e *scripts* embutidos.

6.1 Atributos de <BODY>

Através de atributos de <BODY>, é possível definir cores para os textos, *links* e para o fundo das páginas, bem como uma imagem de fundo (marca d'água):

```
<body bgcolor="#rrggbb" text="#rrggbb" link="#rrggbb"
alink="#rrggbb" vlink="#rrggbb" background="url">
```

Onde:

BGCOLOR: *Background Color*, ou seja, cor de fundo da página. Quando não é indicada a cor de fundo, o *browser* irá mostrar uma cor padrão, geralmente cinza ou branco.

TEXT: Cor dos textos da página (padrão: preto).

LINK: Cor dos *links* (padrão: azul).

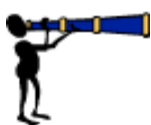
ALINK: Cor dos *links* quando acionados, clicados (padrão: vermelho).

VLINK: Cor dos *links* depois de visitados (padrão: azul escuro ou roxo).

Os valores dos atributos são dados em hexadecimal, equivalentes a cores no padrão RGB (Red, Green, Blue). Existem tabelas de cores com esses valores, mas grande parte dos editores já oferece uma interface bem amigável através da qual é possível escolher as cores desejadas, sem se preocupar com números esdrúxulos tais como **#FF80A0**.

Browsers que seguem a definição de HTML 3.2 em diante, também aceitam **16 nomes de cores**, tirados da paleta VGA do Windows - por exemplo, podemos escrever **BGCOLOR="BLUE"**. Porém, *browsers* mais antigos não apresentarão as cores indicadas.

BACKGROUND: Indica o URL³ da imagem a ser replicada no fundo da página, como uma marca d'água. Para efeitos de design, é possível fixar a imagem de fundo, para que ela não se mova junto com o texto ao se rolar a página. Esse efeito *não é padrão* e funciona no Internet Explorer.



Dica: Os nomes das cores - os 16 nomes de cores aceitos desde a versão 3.2 da HTML são estes: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.

³URL: O sistema de endereçamento da Web é baseado em uma sintaxe chamada URI (*Universal Resource Identifier* - Identificador Universal de Recursos). Os endereços que utilizamos atualmente são os URLs, que seguem essa sintaxe. URL significa *Uniform Resource Locator* - Localizador Uniforme de Recursos. Um exemplo de URL é: <http://www.faccat.br> Esse endereço identifica: o **protocolo de acesso** ao recurso desejado (http) e a **máquina** a ser acessada (www.faccat.br).

7 Fontes

Atributos das fontes:

Cores

As cores são introduzidas através do elemento ``, usando o sistema RGB para cores (da mesma forma que vimos para cores de documentos), como mostrado no exemplo a seguir:

```
<font color="red"> Texto </font>
```

Produzirá na página: a palavra **Texto** escrita na cor **vermelha**.

Assim, um trecho de texto pode ter uma cor diferente da definição geral de cores, feita através dos atributos de `<BODY>`.

Tamanhos

Veja a formatação a seguir: produzirá na página: a palavra **Texto** como o **tamaho 3**.

```
<font size=3> texto </font>
```

Este comando permite que o autor do documento altere o tamanho das letras em trechos específicos de texto. O tamanho básico dos textos é 3 (padrão). Pode-se indicar tamanhos relativos a esse, por exemplo:

```
<font size=+2> Letra maior </font>
```

Letra normal

```
<font size=-2> Letra menor </font>
```

Na página web apareceria assim :

Letra maior Letra normal Letra menor

Tipos de Letra

Uma evolução que permite a escolha da fonte para os textos (tipo de letra), é o atributo FACE:

```
<font face="arial"> Texto </font>
```

Produzirá: a palavra **Texto** com a fonte (tipo de letra) Arial.

Mais exemplos:

```
<font face="verdana" color="blue"> Fonte Verdana Azul </font>
```

Fonte Verdana Azul

```
<font face="arial" color="green"> Fonte Arial Verde </font>
```

Fonte Arial verde

```
<font face="courier new" color="red"> Fonte Courier New Vermelha  
</font>
```

Fonte Courier New Vermelha

8 Cabeçalhos

Há seis níveis de cabeçalhos em HTML, de **<H1>** a **<H6>**, que servem para colocar títulos e subtítulos nas páginas web:

<H1> Este é um cabeçalho de nível 1 **</H1>**

<H2> Este é um cabeçalho de nível 2 **</H2>**

<H3> Este é um cabeçalho de nível 3 **</H3>**

<H4> Este é um cabeçalho de nível 4 **</H4>**

<H5> Este é um cabeçalho de nível 5 **</H5>**

<H6> Este é um cabeçalho de nível 6 **</H6>**

Esses cabeçalhos são mostrados no *browser* da seguinte forma:

Este é um cabeçalho de nível 1

Este é um cabeçalho de nível 2

Este é um cabeçalho de nível 3

Este é um cabeçalho de nível 4

Este é um cabeçalho de nível 5

Este é um cabeçalho de nível 6

8.1 Aninhamento de cabeçalhos

Os cabeçalhos não podem ser aninhados, isto é, a formatação:

<H2>Este é **<H1>**um cabeçalho de nível 1**</H1>** dentro de um cabeçalho de nível 2**</H2>**

→ Pode produzir algum resultado próximo ao desejado:

Este é
um cabeçalho de nível 1
dentro de um cabeçalho de nível 2

Mas o mais comum é que os *browsers* "entendam" essa formatação como sendo:

<H2>Este é**</H2>** **<H1>**um cabeçalho de nível 1**</H1>** dentro de um cabeçalho de nível 2**</H2>**

Ou seja, como se estivesse faltando uma etiqueta de fechamento de <H2> antes de <H1> e faltando uma abertura de <H2> depois do fechamento de <H1>, oferecendo o seguinte resultado:

Este é

um cabeçalho de nível 1

dentro de um cabeçalho de nível 2

Os editores WYSIWYG naturalmente não permitem o aninhamento de cabeçalhos.

8.2 Alinhamento de cabeçalhos

Os cabeçalhos têm atributos de alinhamento:

`<h2 align=center> Cabeçalho Centralizado </h2>`

Cabeçalho centralizado

`<h3 align=right> Cabeçalho alinhado à direita </h3>`

Cabeçalho alinhado à direita

`<h4 align=left> Cabeçalho alinhado à esquerda </h4>`

Cabeçalho alinhado à esquerda (*default* = padrão)

9 Separadores

As quebras de linha do texto fonte não são significativas na apresentação de documentos em HTML, ou seja, ao dar Enter no código fonte da página web nos comandos HTML, o resultado da página no navegador não é alterado, o navegador não reconhece o Enter como quebra de linha. Para organizar os textos são necessários separadores, apresentados a seguir.

9.1 Quebra de linha

Quando deseja-se mudar de linha, deve-se usar o elemento **
**. Isso só é necessário quando precisa-se de uma quebra de linha em determinado ponto, pois os *browsers* já quebram as linhas automaticamente para apresentar os textos. Vem da palavra *break*.

Com sucessivas etiquetas **
**, pode-se inserir diversas linhas em branco nos documentos. Esse elemento tem um atributo especial, que será apresentado no item sobre inserção de imagens.

9.2 Parágrafos

Para separar blocos de texto, usa-se o elemento **<P>**, por exemplo:

```
<p>Parágrafo 1;</p> <p>Parágrafo 2.</p>
```

Que produz o seguinte resultado na página web:

Parágrafo1;

Parágrafo2.

Combinando parágrafos e quebras de linha, tem-se, por exemplo:

```
Parágrafo 1;<br> linha 1 do parágrafo 1, <br>linha 2 do parágrafo 1.<p>Parágrafo 2;<br> linha 1 do parágrafo 2, <br>linha 2 do parágrafo 2.
```

→ O resultado da marcação acima é:

Parágrafo 1;
linha 1 do parágrafo 1,
linha 2 do parágrafo 1.

Parágrafo 2;
linha 1 do parágrafo 2,
linha 2 do parágrafo 2.

<P> tem atributo de alinhamento, semelhante aos cabeçalhos, como nos exemplos a seguir:

```
<p align=center> Assim como os trens, as boas ideias às vezes chegam  
com atraso. <br> (Giovani Guareschi) </p>
```

**Assim como os trens, as boas ideias às vezes chegam com atraso.
(Giovani Guareschi)**

```
<p align=right> Como diz o provérbio chinês: “É melhor passar por  
ignorante uma vez do que permanecer ignorante para sempre”. </p>
```

**Como diz o provérbio chinês: “É melhor passar por ignorante uma vez do que permanecer
ignorante para sempre”.**

```
<p align=left> Este é o alinhamento padrão (default) e, por isso,  
não vou colocar nenhuma frase especial.</p>
```

Este é o alinhamento padrão (default) e, por isso, não vou colocar nenhuma frase especial.

10 Linha Horizontal

`<hr>` insere uma linha horizontal, como a apresentada abaixo:



Essa linha tem diversos atributos, oferecendo resultados diversos:

`<hr size=10>` insere uma linha de largura 10 (pixels):



`<hr width=50%>` insere uma linha que ocupa 50% do espaço horizontal disponível:



`<hr width=30% align=right noshade>` insere uma linha de comprimento 30% (do espaço horizontal disponível), alinhada à direita, sem efeito tridimensional:



`<hr size=70 width=2 align=left>` insere uma linha de largura 70 (pixels), comprimento 2 (pixels), alinhada à esquerda (o Netscape, aparentemente, não aceita esta formatação de `<HR>`):



Dica: Divisões - outros elementos usados para separar porções de texto são o `<DIV>` e o `<CENTER>`. `<CENTER>` centraliza os elementos (textos, imagens, tabelas) que estiverem dentro de sua marcação. `<DIV>` marca uma divisão lógica de um documento e é uma formatação bastante usada atualmente.

11 Comentários

Para fazer algum comentário no código fonte HTML usa-se a seguinte notação:

`<!-- blá blá blá -->`

12 Listas em HTML

Há dois tipos de listas em HTML, apresentadas a seguir:

12.1 Listas com Marcadores

São equivalentes às listas com marcadores do MS-Word, por exemplo:

 item de uma lista

 item de uma lista, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto

 item

- item de uma lista
- item de uma lista, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto
- item

A diferença entre o resultado da marcação HTML e de um Editor de Textos está na mudança dos marcadores, assinalando os diversos níveis de listas compostas:

 Documentos básicos

 Documentos avançados

 formulários

 CGI

 contadores

 relógios

 Detalhes sobre imagens

- Documentos básicos
- Documentos avançados
 - formulários
 - CGI
 - contadores
 - relógios
- Detalhes sobre imagens

12.2 Listas Numeradas

``

`` item de uma lista numerada

`` item de uma lista numerada, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto

`` item de lista numerada

``

- 1. item de uma lista numerada**
- 2. item de uma lista numerada, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto**
- 3. item de lista numerada**

Estas listas não apresentam numeração em formato 1.1, 1.2, etc., quando compostas apresentam-se da seguinte forma:

- 1. Documentos básicos**
- 2. Documentos avançados**
 - 1. formulários**
 - 1. CGI**
 - 2. contadores**
 - 3. relógios**
 - 2. Detalhes sobre imagens**

Porém, através do atributo TYPE (HTML 3.2), pode-se lidar com a numeração dos itens:

`<OL TYPE=I>`

`` Documentos básicos

`` Documentos avançados

`<OL TYPE=a>`

`` formulários

`<OL TYPE=i>`

`` CGI

``

`` contadores

`` relógios

``

`` Detalhes sobre imagens

``

- I. Documentos básicos**
- II. Documentos avançados**
 - a. formulários**
 - i. CGI**
 - b. contadores**
 - c. relógios**
- III. Detalhes sobre imagens**

Ainda segundo HTML 3.2, o atributo START pode indicar o início da numeração da lista:

```
<OL START=4 TYPE=A>
  <LI> um item
  <LI> outro item
  <LI> mais um item
</OL>
```

- D. um item
- E. outro item
- F. mais um item

12.3 Listas e sub-listas

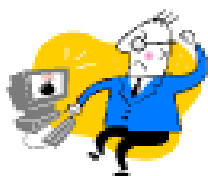
As listas podem ser aninhadas. Por exemplo:

```
<DL>
  <DT> termo a ser definido
  <DD> definição
  <OL>
    <LI> item de uma lista numerada
    <LI> item de uma lista numerada
  </OL>
  <UL>
    <LI> item de uma lista
  </UL>
  <LI> item de uma lista numerada
</DL>
```

termo a ser definido
definição

1. item de uma lista numerada
2. item de uma lista numerada
 - item de uma lista
3. item de uma lista numerada

termo a ser definido
definição



Dica: Se você não está vendo diferença alguma entre as listas comuns e as que têm atributos TYPE, isso se deve ao fato de seu *browser* não estar reconhecendo esses atributos como válidos. Trata-se de um *browser* de versão antiga. Isso deve ser pensado ao usar atributos mais recentes: **nem todo usuário poderá ver o resultado das novas marcações.**

13 Formatação de Frases

Quando disponível no **browser**, é mostrado em **negrito** (em alguns *browsers*, pode aparecer sublinhado).

<i>

Itálico

<u>

Sublinhado. Deve ser usado com cuidado, pois confunde-se com a apresentação de *links*.

<strike> ou **<s>**

Frase riscada.

<big>

Fonte um pouco maior.

<small>

Fonte um pouco menor.

<sub>

Frase em estilo índice, como em **H₂O**, por exemplo.

<sup>

Frase em estilo ^{expoente}, como em **Km²**, por exemplo.

14 Marquee

É possível obter o efeito de animação de texto, através da formatação **<marquee>**.

<marquee behavior = efeito> Texto **</marquee>**

Atributos de largura e direção do efeito permitem diversas apresentações diferentes, por exemplo:

<marquee behavior=scroll width=30%> Texto **</marquee>**

<marquee behavior=slide direction=right> Texto **</marquee>**

<marquee behavior=slide direction=left> Texto **</marquee>**

Atenção: O efeito Marquee só é executado no Internet Explorer e em versões recentes do Netscape - e de maneiras diferentes. Comando muito antigo e fora de uso!!!!

15 Ligações (uso de Links)

Com HTML é possível fazer ligações de uma região de texto (ou imagem) a um outro documento (ou a outra parte do próprio documento). Você já deve ter visto em alguma página internet exemplos dessas ligações: o *browser* destaca essas regiões e imagens do texto, indicando que são ligações de hipertexto - também chamadas *hypertext links* ou *hyperlinks* ou simplesmente *links*, onde normalmente, o mouse vira uma “mãozinha” e ao clicar “chama-se” (abre-se) um outro documento, página web ou figura por exemplo.

Para inserir um link em um documento, utiliza-se a etiqueta `<a>`, da seguinte forma:

```
<a href = "arquivo_destino"> âncora </a>
```

Onde:

arquivo_destino: é o endereço do documento de destino, da página ou imagem a qual deseja-se abrir.

âncora: é o texto ou imagem que servirá de ligação hipertexto do documento sendo apresentado para o documento de destino.

15.1 Atributos

`<a>` tem vários atributos que são utilizados de acordo com a ação associada ao link. Os mais usados são apresentados a seguir:

HREF: Indica o arquivo de destino da ligação de hipertexto.

TARGET: Indica o quadro (*frame*) em que será carregado o arquivo_destino. Exemplo:
`target="_blank"`

NAME: Marca um **indicador**, isto é, uma região de um documento como destino de uma ligação.
Maiores detalhes na seção sobre indicadores.

15.2 Caminhos

Os links podem estar indicados como caminhos relativos ou absolutos.

→ Caminho Relativo

O caminho relativo pode ser usado sempre que for necessário **fazer referência a um documento armazenado no mesmo servidor do documento atual**.

Através do campo de endereços do *browser*, é possível identificar se um documento (página) que está sendo visualizado está dentro de algum diretório (pasta). Como por exemplo, se estivermos em um *browser* acessando a página da FACCAT e vemos o seguinte endereço: *http://www.faccat.br/apostilas/apostila_de_informatica.html*. O que pode-se concluir é que o documento que está sendo visualizado no momento, chamado **apostila_de_informatica.html**, está localizado dentro de um diretório (pasta) chamado **apostilas** do servidor **www.faccat.br**.

Então, para escrever, por exemplo um link deste documento (*apostila_de_informatica.html*) para um documento chamado **doc.html** que está localizado no diretório **/apostilas/documentos/** do mesmo servidor **www.faccat.br**, é necessário escrever:

`` exemplo de caminho relativo ``

Para usar *links* com caminhos relativos é preciso, portanto, conhecer a estrutura do diretório do servidor no qual se está trabalhando, pois deve-se indicar todo o caminho onde está o documento no qual está se referindo no *link*. Quando há alguma dúvida, o melhor é usar o caminho absoluto.

→ Caminho Absoluto

Utiliza-se o caminho absoluto quando se deseja referenciar um documento que esteja em outro servidor, por exemplo:

`` FACCAT - Faculdades de Taquara ``

Que oferece um *link* **FACCAT - Faculdades de Taquara** que ao ser clicado com o mouse abrirá a página cujo endereço é <http://www.faccat.br>.

Com a mesma sintaxe, é possível escrever *links* para qualquer servidor de informações da Internet.

15.3 Indicadores

O atributo **NAME** permite indicar um trecho de documento como **ponto de chegada** de uma ligação hipertexto (link). A formatação:

```
<a name="inicio"> Indicadores (uso de links) </a>
```

faz com que a âncora **Indicadores (uso de links)** seja o destino de um link.

Ao escrever:

```
<a href="#inicio"> Topo do Documento </a>
```

tem-se uma ligação hipertexto para um trecho deste mesmo documento.

16 Inserção de Imagens

O elemento **IMG** insere imagens que são apresentadas junto com os textos. Um atributo **SRC** deve estar presente, da seguinte forma:

```

```

Onde **URL_imagem** é o URL do arquivo que contém a imagem que se quer inserir; pode ser referenciada uma imagem que esteja em um outro servidor (o que, logicamente, **não é conveniente**).

Assim, escrevendo:

```
<img src = "nome_da_figura.extensão">
```

ou, se ela estiver dentro de uma pasta no mesmo servidor:

```
<img src = "nome_da_pasta/nome_da_figura.extensão">
```

Por exemplo, se deseja-se inserir na página uma imagem chamada **arvore.gif** localizada no mesmo servidor e na mesma pasta, a *tag* seria assim:

```
<img src = "arvore.gif">
```

As imagens usadas na Web, normalmente são armazenadas em arquivos com as seguintes extensões: **.gif**, **.jpg**, **.png**, **.bmp**.

16.1 Atributos Básicos de Imagem

ALT: indica um texto alternativo, descrevendo brevemente a imagem, que é apresentado no lugar da imagem nos *browsers*. Ou quando se desabilita o carregamento de imagens em *browsers* gráficos. É recomendável que esteja sempre presente.

TITLE: é o que aparecerá ao “passar o mouse sobre a imagem”.

```

```

Dessa forma:

```

```

É apresentado nos *browsers* gráficos assim: **New!** e, nos *browsers* texto, assim: [Novo!!!]. E ao passar o mouse sobre a figura aparecerá a descrição Novo Vermelho!!!

WIDTH e HEIGHT

Atributos de dimensão da imagem, em *pixels*⁴. Grande parte dos Editores HTML coloca automaticamente os valores destes atributos, quando indica-se a inserção de uma imagem.

```

```

Uma das vantagens de usar esses atributos é que o *browser* pode montar mais rapidamente as páginas, por saber de antemão o espaço que deverá ser reservado a elas.

BORDER

Quando uma frase é marcada como âncora de um link, ela se apresenta sublinhada. Quando uma imagem faz as vezes de âncora, ganha uma borda que indica sua condição de link.

Porém, por questões de apresentação, nem sempre interessa ter essa borda ao redor da imagem. Assim, com o atributo **BORDER**, é possível controlar esse detalhe.

* Se quiser uma borda mais larga:

```
<a href="URL">  </a>
```

* Se quiser uma imagem sem borda:

```
<a href="URL">  </a>
```

Essa borda pode ser apresentada também em imagens que não são âncora de *links*. Basta aplicar, por exemplo, a formatação:

```

```

Assim, é possível dar mais destaque a uma imagem, sem ser necessário editá-la:



Foto Original



Foto com Borda gerada pelo HTML

⁴ *Pixel*: O pixel, ou ponto, é a menor unidade que compõe uma imagem digital, ele contém os atributos de cor de cada ponto, a maior ou menor quantidade de *pixels* em uma área é que determinam a resolução da imagem, bem como seu maior ou menor tamanho em *bytes*. São pontos por centímetro quadrado. Quanto maior o número de *pixels* maior a qualidade da imagem e maior será o tamanho do arquivo também.

ALIGN

```

```

Existem também atributos de alinhamento, que produzem os seguintes resultados:

ALIGN=TOP: Alinha o texto adjacente com o topo da imagem, embora com linhas compridas o resultado não seja muito bom.

ALIGN=MIDDLE: Alinha o texto adjacente com o meio da imagem, embora com linhas compridas o resultado não seja muito bom.

ALIGN=BOTTOM: Alinha o texto adjacente com a parte de baixo da imagem (default).

ALIGN=RIGHT: Alinha imagem à direita, e tudo o que houver ao redor (texto, outras imagens) a partir do topo da imagem.

ALIGN=LEFT: Alinha imagem à esquerda, e tudo o que houver ao redor (texto, outras imagens) a partir do topo da imagem.

→ Para ter duas imagens, uma em cada margem, numa mesma linha, escreva:

```
 ...e pode-se escrever à vontade  
entre as imagens!
```

Isto resulta em:



...e pode-se escrever à vontade entre as imagens!



Um detalhe surgido com o alinhamento de imagens foi a necessidade de se liberar o texto desse alinhamento, ou seja: suponha um texto mais ou menos curto, que deseja-se colocar com a imagem ilustrativa, mas gostaria que este trecho já estivesse abaixo da imagem! De acordo com o comprimento da primeira frase, não seria possível usar o alinhamento TOP.

Para conseguir isso, seria necessário incluir diversos
 consecutivos, inserindo linhas em branco; mesmo assim, o resultado final poderia ser bem pouco elegante. Surgiu, então, o atributo CLEAR para
 .

Com esse atributo, é possível, por exemplo ter um texto posicionado no ponto em que a margem direita fica livre, com <BR CLEAR=RIGHT> ou no ponto em que a margem esquerda fica livre, com <BR CLEAR=LEFT> . Dessa maneira, pode-se controlar bem a posição relativa dos textos. Também pode-se posicionar o texto no ponto em que ambas as margens estão livres. Isso é conseguido com <BR CLEAR=ALL> .

17 Molduras de Imagem

Para melhorar ainda mais a apresentação das imagens junto com os textos, foram desenvolvidos atributos de moldura. Estes atributos definem o espaço - vertical e horizontal - deixado entre as imagens e os textos circundantes:

```

```

```

```

Outros exemplos:

```

```

```

```

Abaixo, um exemplo com os dois atributos, através da formatação:

```

```

18 Tabelas

A formatação de tabelas foi adotada bem antes de sua inclusão na definição de HTML. A manipulação de tabelas, mesmo em editores, é trabalhosa. A maior diferença entre tabelas em HTML e em editores como o MS-Word, entretanto, é o fato das tabelas em HTML serem definidas apenas **em termos de linhas e não de colunas**. Mas isso será percebido no decorrer deste capítulo.

As tabelas por muito tempo foram utilizadas para fazer os layouts das páginas web. Hoje em dia não se deve usar tabelas para layout, ou seja, usa-se tabelas em HTML apenas se for necessário ter uma tabela na página. Pesquise sobre o conceito de *tableless*.

18.1 Elementos básicos de tabelas

<TABLE>...</TABLE> delimita uma tabela. Um atributo básico é **BORDER**, que indica a apresentação da borda.

```
<table border="borda">
...
</table>
```

18.2 Títulos, linhas e elementos

<CAPTION>...</CAPTION>
define o título da tabela

<TR>...</TR>
delimita uma linha

<TH>...</TH>
define um cabeçalho para colunas ou linhas (dentro de <TR>)

<TD>...</TD>
delimita um elemento ou célula (dentro de <TR>)

18.3 Uma tabela simples

```
<table border=4>
  <caption>Primeiro Exemplo</caption>
  <tr><th>Coluna 1</th> <th>Coluna 2</th></tr>
  <tr><td>Linha 1, Coluna 1</td> <td>Linha 1, Coluna 2</td></tr>
  <tr><td>Linha 2, Coluna 1</td> <td>Linha 2, Coluna 2</td></tr>
</table>
```

Primeiro Exemplo

Coluna 1	Coluna 2
Linha 1, Coluna 1	Linha 1, Coluna 2
Linha 2, Coluna 1	Linha 2, Coluna 2

18.4 Títulos compreendendo mais de uma coluna ou linha

É possível englobar colunas e linhas, através dos atributos **COLSPAN** (para colunas) e **ROWSPAN** (para linhas):

```
<table border=1>
  <tr><th colspan=2>colunas 1 e 2</th></tr>
  <tr><td>linha1, coluna 1</td><td>linha 1, coluna 2</td></tr>
  <tr><td>linha 2, coluna 1</td><td>linha 2, coluna 2</td></tr>
  <tr><th rowspan=3>3 linhas</th><td>uma linha</td></tr>
  <tr><td>duas linhas</td></tr>
  <tr><td>tres linhas</td></tr>
</table>
```

Colunas 1 e 2	
linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2
3 linhas	uma linha
	duas linhas
	tres linhas

Neste exemplo, vemos que o cabeçalho **Colunas 1 e 2** compreende duas colunas (**COLSPAN=2**); o cabeçalho **3 linhas** compreende, por sua vez, 3 linhas (**ROWSPAN=3**).

18.5 Tabelas sem borda

As páginas deste tutorial foram construídas com tabelas sem borda. Para tanto, foi empregada a seguinte declaração:

```
<table border="0">
...
</table>
```



Dica: Pior que os acentos... A formatação de tabelas é complicada, e o texto fonte chega a ser quase ininteligível quando montamos tabelas complexas e fazemos uso de seus diversos atributos. A melhor opção, sem dúvida, é usar os editores WYSIWYG. Alguns editores de modo texto têm uma interface gráfica que ajuda na criação de tabelas, mas a edição de tabelas já existentes precisa ser feita à mão.

18.6 Extensões de Tabelas

Diversas extensões de tabelas possibilitam a apresentação de efeitos muito bons nas páginas.

Cor de fundo

```
<table border=5 cellpadding=10>
  <tr>
    <td bgcolor="#e1ffd9">janeiro</td>
    <td>fevereiro</td>
    <td bgcolor="#e1ffd9">março</td>
  </tr>
  <tr>
    <td>abril</td>
    <td bgcolor="#e1ffd9">maio</td>
    <td>junho</td>
  </tr>
</table>
```

janeiro	fevereiro	março
Abril	maio	junho

Cor de borda

```
<table border=5 cellspacing=5 cellpadding=10 bgcolor="#e1ffd9"
bordercolor="#00ff00">
```

janeiro	fevereiro	março
abril	maio	junho

Exemplo colocando cor de fundo e na borda:

```
<table border="1" cellspacing="0" cellpadding=10
bordercolor="#00ff00">
```

```
  <tr>
    <td bgcolor="#e1ffd9">janeiro</td>
    <td>fevereiro</td>
    <td bgcolor="#e1ffd9">março</td>
  </tr>
  <tr>
    <td>abril</td>
    <td bgcolor="#e1ffd9">maio</td>
    <td>junho</td>
  </tr>
</table>
```

janeiro	fevereiro	março
abril	maio	junho

19 Formulários HTML

Os formulários HTML são estruturas que permitem que usuários submetam dados a uma página. Esses dados podem ser tratados e/ou armazenados dependendo da aplicação.

19.1 Criando um Form

Os **forms** fazem parte do código HTML, portanto devem ser definidos como as tags desta linguagem. Sempre com uma tag de abertura e outra de fechamento, inserindo o conteúdo entre estas duas.

Exemplo:

```
<form action="pagina.php" method="post"> //tag de abertura  
  
//conteúdo do form  
  
</form> //tag de encerramento
```

19.2 Parâmetros do Form

Action: é o script ou página para onde os dados serão submetidos. Neste script que normalmente os dados são tratados.

Method: é o método de envio dos dados. Pode ter dois valores:

GET: passa os valores pela URL, ou seja, podemos ver as variáveis passadas na Url da página destino definida no campo action. Não é muito aconselhável o uso do método GET, pois ele expõe o nome e valor das variáveis, o que pode ser utilizado por pessoas com intenções maléficas.

POST: passa as variáveis de maneira transparente para o usuário. É o método mais aconselhável.

Veja mais diferenças entre Get e Post em:

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/Sending_and_retrieving_form_data

19.3 Elementos do Form

Pode-se inserir vários tipos de entrada de dados em um formulário, a maioria delas definida pela tag **input**. Todo elemento possui um parâmetro **name** que é utilizado para identificar a variável onde o dado está contido no script destino. Lembrando que estes elementos devem ser inseridos entre as tags `<form>` e `</form>`.

CAMPO TEXTO: campo para entrada de texto comum.

Declaração:

```
<input type="text" size="" maxlength="" name="" value="">
```

Parâmetros:

Size: tamanho do elemento em caracteres na página HTML (que será exibido na tela).

Maxlength: tamanho máximo do texto que pode ser inserido no elemento.

Value: utilizado quando há necessidade de se pré-definir um valor para o elemento. Este valor pode ser normalmente alterado pelo usuário.

CAMPO SENHA: tipo de campo idêntico ao anterior, mas quando o usuário digita, os caracteres são substituídos por “*”. **Atenção:** o campo senha não possui nenhum tipo de criptografia, apenas coloca uma máscara no texto inserido.

Declaração:

```
<input type="password" size="" maxlength="" name="" value="">
```

Parâmetros:

Size: tamanho do elemento em caracteres na página HTML (que será exibido na tela).

Maxlength: tamanho máximo do texto que pode ser inserido no elemento.

Value: utilizado quando há necessidade de se pré-definir um valor para o elemento. Este valor pode ser normalmente alterado pelo usuário.

BOTÃO RÁDIO (RADIO BUTTON): utilizado para entradas de múltipla escolha onde o usuário só pode escolher uma única opção. Para que o interpretador saiba que as opções fazem parte do mesmo grupo, e permita que só uma seja selecionada, basta *colocar o mesmo nome no parâmetro **name*** dos botões rádio.

Declaração:

```
<input type="radio" name="" value="" checked>
```

Parâmetros:

Checked: se for declarado, o elemento terá seu estado inicial como selecionado.

Value: valor que será passado à página destino quando o formulário é submetido, se este elemento estiver selecionado.

BOTÃO DE CHECAGEM (CHECK BOX): utilizado para entradas de múltipla escolha onde o usuário pode escolher várias opções. *Cada opção deve ter um nome independente.*

Declaração:

```
<input type="checkbox" name="" value="" checked>
```

Parâmetros:

Checked: se for declarado, o elemento terá seu estado inicial como marcado.

Value: valor que será passado à página destino quando o formulário é submetido se este elemento estiver marcado.

BOTÃO SUBMETER (SUBMIT): botão que submete o formulário à página destino especificada no parâmetro **action** do `form`.

Declaração:

```
<input type="submit" name="" value="">
```

Parâmetros:

Value: texto que aparecerá no botão.

BOTÃO RESET: volta todos os campos do formulário para os valores iniciais, ou seja, os valores especificados nos parâmetros **value** de cada um dos campos.

Declaração:

```
<input type="reset" name="" value="">
```

Parâmetros:

Value: texto que aparecerá no botão.

SELECT: utilizado para selecionar uma ou mais opções de uma lista predefinida.

Declaração:

```
<select name="" size="" multiple>
  <option value="">texto da opcao</option>
</select>
```

Parâmetros:

Multiple: se existir, permite que sejam selecionadas múltiplas opções através das teclas **Ctrl** ou **Shift**.

Size: número de opções exibidas por vez na tela. Se o **size** estiver setado para "1" que é o *default* e não existir o parâmetro **multiple**, o elemento é exibido como um **Combo Box**. Caso contrário, é exibido como um **Select List**.

Option: cada subtag **option** adiciona uma opção ao elemento.

Value: o **value** de cada **option** é o valor passado caso aquela opção seja selecionada.

Observação: O texto da opção deve ser especificado entre as tags **<option>** e **</option>**

ÁREA DE TEXTO: permite a entrada de um texto com várias linhas.

Declaração:

```
<textarea cols="" rows="">texto</textarea>
```

Parâmetros:

Cols: número e caracteres por linha.

Rows: número de linhas do campo.

Observação: não existe o parâmetro **value**, o texto inicial deve ser definido entre as tags **<textarea>** e **</textarea>**.

19.4 Trabalhando com os dados enviados pelo Form

Como visto anteriormente, os **forms** submetem os dados à um script ou página especificada no parâmetro **action**. Este script ou página pode ser ou não o mesmo onde o form está contido. No script destino, são criadas automaticamente variáveis com os mesmos nomes dos elementos do form, contendo o valor submetido.

Por exemplo:

Em uma página qualquer tem-se o seguinte código:

```
<form action="teste.php" method="post">
  Digite seu nome <input type="text" name="nome"><br>
  <input type="submit" name="enviar" value="Enviar">
</form>
```

Que resulta no seguinte formulário:

Digite seu nome

Quando este form for submetido (clicando no botão Enviar), na página teste.php será criada uma variável com o nome **\$nome** contendo o texto que foi digitado no campo.

19.5 Colocando em Prática

Vamos criar duas páginas (dois arquivos) no nosso servidor de teste:

tutorial.php: será o formulário com algumas perguntas.

resultado.php: onde serão exibidas as respostas de uma maneira amigável.

No primeiro arquivo, tutorial.php, você deve criar uma estrutura HTML normal, e dentro do body criar um formulário que submete os dados lidos no formulário para a página resultado.php, onde essas informações serão trabalhadas, da seguinte maneira:

Criar um campo texto para entrada do nome, um checkbox perguntando se o usuário leu o tutorial e um campo select para entrada da nota dada ao tutorial. Salvar o arquivo com o nome **tutorial.php**, conforme código abaixo:

```
tutorial.php x resultado.php x
1 <html>
2 <head>
3     <meta charset="utf-8">
4     <title>Meu Formulário</title>
5 </head>
6 <body>
7     <h3 align=center>Meu Primeiro Formulário HTML</h3>
8     <form action="resultado.php" method="post">
9         Nome <input type="text" name="nome"><br>
10        Eu li o tutorial todinho <input type="checkbox" name="leu"><br>
11        Qual nota você dá para o tutorial?
12        <select name="nota">
13            <option value="0">Nota 0</option>
14            <option value="1">Nota 1</option>
15            <option value="2">Nota 2</option>
16            <option value="3">Nota 3</option>
17        </select>
18        <input type="submit" value="Enviar">
19    </form>
20 </body>
21 </html>
```

A página deve ficar da seguinte maneira (ao abrir no browser):

Meu Primeiro Formulário HTML

Nome

Eu li o tutorial todinho ☐

Qual nota você dá para o tutorial? Nota 0 ▾ Enviar

Agora você deve criar o arquivo **resultado.php**, que receberá os dados enviados pelo formulário quando o usuário clicar no botão Enviar. Da seguinte maneira:

```
tutorial.php x resultado.php x
1 <html>
2 <head>
3     <meta charset="utf-8">
4     <title>Resultado</title>
5 </head>
6 <body>
7     <h3 align=center>Dados Preenchidos no Formulário</h3>
8     Usuário: <?php echo $_POST['nome']; ?><br>
9     Leu o tutorial:
10    <?php
11        if (isset($_POST['leu'])) /*isset testa se a variável existe*/
12        {
13            echo "Sim, leu!";
14        }
15        else
16        {
17            echo "Não leu.";
18        }
19    ?><br>
20    Nota: <?php echo $_POST['nota']; ?>
21 </body>
22 </html>
```

Para testar o formulário é necessário ter instalado o **Wamp** e ter salvo os dois arquivos (tutorial.php e resultado.php) **dentro da pasta www do Wamp**. Também verifique se os serviços do Wamp estão **online** (o ícone deve estar verde). Você deve abrir o arquivo tutorial.php em um navegador da seguinte forma: **localhost/tutorial.php**, pois só assim os comandos PHP irão funcionar. No Mozilla Firefox, aparecerá da seguinte maneira:



Meu Primeiro Formulário HTML

Nome

Eu li o tutorial todinho ☐

Qual nota você dá para o tutorial?

Conclusão: este tutorial mostrou como utilizar `forms` HTML, explicando detalhadamente seus elementos e respectivos parâmetros e como capturar e utilizar os dados enviados pelos mesmos usando a linguagem PHP e o pacote Wamp.

Material completo com exemplos e detalhes sobre HTML que serviu como uma das fontes para elaboração desta apostila:

<http://www.icmc.usp.br/ensino/material/html/intro.html>

Autora: **Maria Alice Soares de Castro**

Também foi usado como fonte:

<http://www.juliobattisti.com.br/tutoriais/juliosabreu/formularioshtml001.asp>